

SQL

- SQL (baca SEQUEL) pertama kali didefinisikan oleh Chamberlin (IBM Research Laboratory, San Jose, California).
- Karena kemudahannya untuk mengakses data, maka SQL cepat menjadi populer, sehingga banyak DBMS yang memiliki fasilitas SQL untuk mengakses data.
- Terdapat berbagai dialek SQL yang berbeda satu dengan yang lain.
- Pada kuliah ini akan dibicarakan dialek SQL yang standard.

Perintah-perintah dalam SQL dapat dikelompokkan menjadi 2 kelompok:

- DDL (Data Definition Language) dan
- DML (Data Manipulation Language)

1. DDL (Data Definition Language)

membuat tabel

merubah tabel

membuat index

menghapus tabel dan index

2. DML (Data Manipulation Language)

tambah data

update data

hapus data

query data

MEMBUAT TABEL

```
CREATE TABLE Nama_tabel  
([NamaKolom Typedata {NOT NULL}]....)
```

Contoh:

```
Create Table Transkrip  
(NO_MHS      INTEGER      NOT NULL,  
NO_MK   CHAR(6) NOT NULL,  
NILAI           SMALLINT);
```

MERUBAH TABEL

```
ALTER TABLE Nama_tabel  
ADD NamaKolom   Typedata
```

Contoh:

```
ALTER TABLE MAHASISWA  
ADD Alamat char(25);
```

MEMBUAT INDEX

```
CREATE {UNIQUE} INDEX NamaIndex  
ON Nama Tabel
```

UNIQUE Jika nilai pada kolom yang diindex terdapat beberapa nilai yang ditampilkan hanyalah satu nilai saja

ASC Urutan dari Kecil-Besar (Naik)

DESC Urutan dari Besar-Kecil (Turun)

Default ASC

Contoh:

```
CREATE INDEX IDXMHS  
ON Mahasiswa (NO_MHS)
```

MENGHAPUS TABEL & INDEX

```
DROP TABLE Nama_tabel;  
DROP INDEX Nama_Index;
```

MENAMBAH DATA KE TABEL

```
INSERT INTO Nama_tabel {[[NamaKolom]...]}
```

VALUES ([konstan]...)

INSERT INTO Nama_tabel {[Namakolom]...}

SELECT ... FROM....

Contoh:

Insert Into Transkrip

(NOMHS, KDMK, NILAI) VALUES (2930,'MSM160',80)

Insert Into Transkrip

VALUES (2930,'MSM160', 80)

Insert Into NewTranskrip

Select * From Transkrip

Where KDMK <> 'MSM 100'

Jumlah kolom dalam insert into = Jumlah kolom dalam
SELECT

Type data kolom-kolom dalam Insert Into = tipe data kolom-
kolom dalam SELECT

MENGHAPUS DATA

DELETE FROM NamaTabel

{WHERE Kondisi}

Contoh:

Delete

From transkrip;

Delete

From transkrip

Where NoMhs = 3001;

MENGUPDATE DATA

UPDATE NamaTabel

SET [NamaKolom =[Ekspresi]]...

{WHERE Kondisi}

Contoh:

UPDATE MataKuliah

Set NamaMK = 'Program Terstruktur'

Where KdMk = 'MSM 200';

UPDATE Transkrip

Set Nilai = Nilai + 10;

Where KDMK = 'MSM 260'

QUERY DATA

SELECT {DISTINCT| ALL} [*| [Ekspresikolom]...]

FROM [Nama Tabel]....

{WHERE Kondisi}

{GROUP By [Nama Kolom]...}

{HAVING Kondisi}}

{ORDER By [URUTAN]....}

Keterangan:

Setiap perintah SELECT minimal harus ada SELECT dan FROM.

WHERE, GROUP BY, HAVING, ORDER BY bersifat optional (boleh ada, boleh tidak)

GROUP By harus sesudah WHERE atau FROM, dan ORDER By harus selalu terakhir.

HAVING hanya dapat digunakan jika ada GROUP By.

Betul/Salah perintah Select berikut ini:

Select

Where

Order by

Select....

From..

Having...

Group By....

Select

Order by.....

From..

Group by.....

URUTAN PEMROSESAN PERINTAH SELECT

FROM	Dari tabel mana data berasal
WHERE	Pilih record yang memenuhi kondisi
GROUP BY	Kelompokkan berdasarkan nilai yang sama dari kolom
HAVING	Pilih kelompok yang memenuhi kondisi
SELECT	Pilih Kolom

ORDER BY urutkan baris berdasarkan kolom

QUERY SEDERHANA

Select *

From mahasiswa;

Select Distinct KdMk

From Transkrip;

PENGUNAAN WHERE

Diikuti oleh:

- Perbandingan sederhana

- Kondisi (syarat) yang berhubungan dengan AND, OR, dan NOT

- Operator BETWEEN

- Operator IN

- Operator LIKE

- Operator IN dengan sub-query

- Operator perbandingan dengan sub-query

- Operator ANY dan ALL

- Operator EXIST

PERBANDINGAN SEDERHANA: menggunakan operator perbandingan

=; < ; > ; <=; >=; <> (!=)

Contoh:

Select NOMHS

From Mahasiswa

Where Kota = 'Jakarta'

Select NOMHS, NamaMhs

From Mahasiswa

Where Year(TglLahir) = 1970

Where 1998 - year(TglLahir) = 25

Kondisi dengan AND, OR, dan NOT

Contoh:

Select NOMHS

From mahasiswa

Where Kota = 'Jakarta'

AND JnKel = 'L'

catatan : operator = pada contoh di atas dapat diganti dengan operator yang lain.

Operator BETWEEN

- Mengecek apakah nilai dari kolom dalam suatu range
- Format: *<kolom>* {NOT} BETWEEN <Nilai> AND <Nilai>

Contoh:

Select NOMHS, NAMAMHS

From Mahasiswa

Where Year(TglLahir) Between 1970 AND 1975

Note:

<kolom> between A AND B \Leftrightarrow (*<kolom>* \geq A) AND
(*<kolom>* \leq B)

<kolom> NOT Between A AND B \Leftrightarrow NOT(*<kolom>* Between
A AND B) \Leftrightarrow (*<kolom>* $<$ A OR (*<kolom>* $>$ B)

Operator IN

Select NoMhs, NamaMhs, Kota From Mahasiswa

Where Kota = "Bandung"

OR Kota = "Jakarta"

OR Kota = "Medan"

OR Kota = "Surabaya"

Dapat diganti dengan menggunakan operator IN, sebagai berikut:

Select NoMhs, NamaMhs, Kota

From Mahasiswa

Where Kota IN ("Bandung","Jakarta","Medan","Surabaya")

Operator LIKE

% : Pengganti 0, 1, atau lebih karakter

_ : underscore pengganti satu karakter

Contoh:

Select NoMhs

From Mahasiswa

Where NamaMhs LIKE '%n'

Contoh Lain:

Nama mata kuliah memuat string 'ta'

Nama mahasiswa panjangnya 10 huruf

Nama mahasiswa minimal panjangnya 10

Operator IN dengan sub-query

Contoh:

Select NoMhs, NamaMhs

From Mahasiswa

Where Kota IN

(Select Kota

From Mahasiswa

Where NoMhs IN (2946, 5280)

Select NoMhs, NamaMhs

From Mahasiswa

Where NoMhs IN

(Select NoMhs

From Transkrip

Where NoMk = 'MSM100')

Operator TETHA dengan sub-query

- Operator tetha dapat digunakan jika sub-query menghasilkan tepat satu nilai

Contoh:

Select NoMhs, NamaMhs

From Mahasiswa

Where Kota =

(Select Kota From Mahasiswa

Where NoMhs = 5830)

Select NoMhs, NamaMhs

From Mahasiswa

Where Year(TglLahir) < (Select Year(TglLahir)

From Mahasiswa

Where NoMhs = 5830

Soal:

Tampilkan Nomor dan nama mahasiswa yang tahun lahirnya sama dengan tahun lahirnya 'Budi', tetapi Budi tidak muncul dalam hasil.

Operator ANY dan ALL

Jika Y1,Y2,Y3,...., dan Yn nilai-nilai yang dihasilkan sub-query S, maka:

NamaKolom < ALL (s) \Leftrightarrow

(NamaKolom < Y1) AND

(NamaKolom < Y2) AND

.....

(NamaKolom < Yn)

Contoh:

Select NoMhs, NamaMhs

From Mahasiswa

Where Year(TglLahir) <= ALL

(Select Year(TglLahir) from Mahasiswa)

Sedangkan

NamaKolom < ANY (S) \Leftrightarrow

(NamaKolom < Y1) OR

(NamaKolom < Y2) OR

.....

(NamaKolom < Yn)

Operator EXIST

Select NamaMhs

From Mahasiswa

Where EXIST

(Select *

From Transkrip

Where NoMhs = Mahasiswa.NoMhs AND

NoMk = 'MSM160')

GROUP BY

Mengelompokkan record berdasarkan nilai suatu atribut (kolom)

Contoh:

Tabel Transkrip

NoMhs	KdMk	Nilai
2530	MSM100	90
2530	MSM200	80
2700	MSM100	60
2700	MSM200	80
2530	MSM300	90

Select NoMhs

From Transkrip

Group by NoMhs

NoMhs	KdMk	Nilai

2530	MSM100,MSM200,M2M300	90,80,90
2700	MSM100,MSM200	60,80

Hasil akhir:

NoMhs

2530

2700

HAVING

- Memilih (menyeleksi) kelompok berdasarkan karakteristik dari kelompok

- Selalu diikuti oleh kelompok fungsi dalam SQL:

COUNT - banyaknya nilai suatu kolom

- banyaknya baris dalam tabel

MIN - Nilai terkecil dalam suatu kolom

MAX- Nilai terbesar dalam suatu kolom

SUM- Nilai total dalam suatu kolom

AVG - Nilai rata-rata dalam suatu kolom

COUNT(*) menghitung jumlah semua baris dalam tabel tanpa menghilangkan duplikasi

COUNT, AVG, SUM dapat menggunakan DISTINCT untuk menghilangkan duplikasi sebelum fungsi tersebut diberlakukan.

Contoh:

```
Select COUNT(*)
```

```
from Mahasiswa
```

```
Select COUNT(distinct NoMhs)
```

```
From transkrip
```

```
Where nilai >= 80
```

```
Select NoMhs, AVG(Nilai)
```

```
From(transkrip)
```

```
Group by NoMhs
```

```
Having AVG(Nilai) > 80
```